

ΕΠΛ 003:
ΕΠΙΣΤΗΜΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΠΛΗΡΟΦΟΡΙΚΑ ΣΥΣΤΗΜΑΤΑ

Δρ. Κουζαπάς Δημήτριος
Πανεπιστήμιο Κύπρου - Τμήμα Πληροφορικής

Λειτουργικά συστήματα

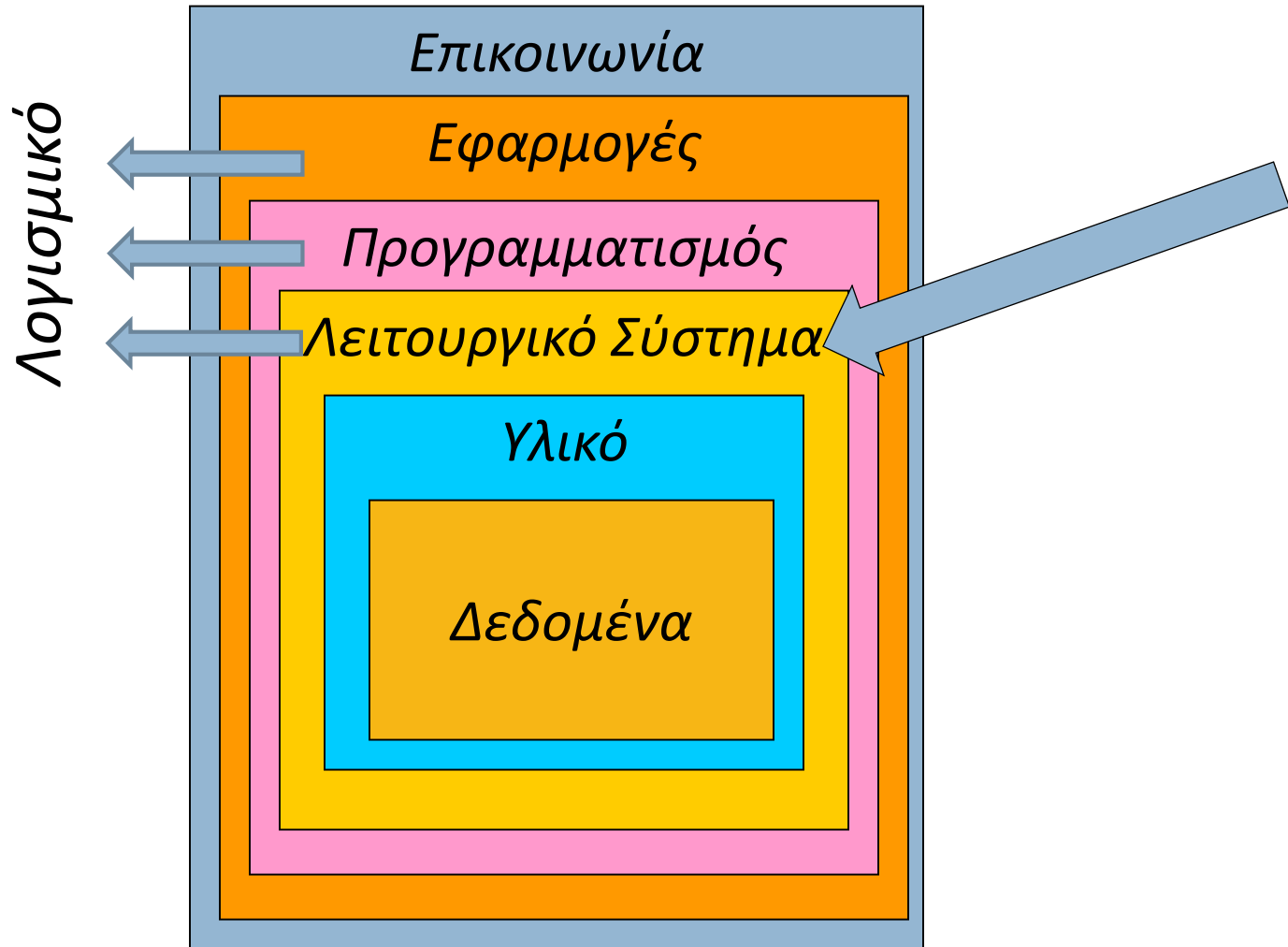
Στόχοι

1

- Να εξηγήσουμε τη διάκριση μεταξύ **Λογισμικού Συστημάτων** και **Λογισμικού Εφαρμογών**.
- Να περιγράψουμε τι είναι τα **Λειτουργικά Συστήματα** και να δούμε σύντομα την ιστορία της εξέλιξής τους.
- Να περιγράψουμε τις σημαντικότερες έννοιες και τεχνικές γύρω από τη **Διαχείριση Μνήμης**.
- Να περιγράψουμε τις βασικές έννοιες σε σχέση με τη **Διαχείριση Διεργασιών**.
- Να περιγράψουμε κάποιους απλούς αλγορίθμους για τον **Χρονοπρογραμματισμό της ΚΜΕ**.

Υπολογιστικά Συστήματα: Στρώματα

2



Λογισμικό



3

- Λογισμικό είναι το σύνολο των **προγραμμάτων** που εκτελεί το Υπολογιστικό Σύστημα:
 - **Λογισμικό Εφαρμογών:** Προγράμματα που **εκτελούν παραγωγικές εργασίες για τους χρήστες**. Καλύπτουν κάποια συγκεκριμένη ανάγκη – Επιλύουν ένα πρόβλημα
 - **Λογισμικό Συστήματος:** Προγράμματα που **εξυπηρετούν τις ανάγκες του Υλικού του Η/Υ και παρέχουν τις αναγκαίες υπηρεσίες που χρειάζεται το Λογισμικό Εφαρμογών από το Υλικό**

Λογισμικό: Δύο Κατηγορίες

4

Γραφικό
Περιβάλλον

Λογισμικό
Εφαρμογών

Ηλεκτρονικό
Ταχυδρομείο

Διαχείριση Βάσης
Δεδομένων

Πρόγραμμα
Ζωγραφικής

Επεξεργαστής
Κειμένου

Πλοηγός
Ιστού

Πυρήνας Λογισμικού Συστήματος

Λογισμικό
Συστημάτων

Οδηγός

Οδηγός

Οδηγός

Οδηγός

Οδηγός

Οδηγός

Οδηγός

Οδηγός



Λογισμικό: Δύο Κατηγορίες

5

Λογισμικό Συστημάτων

π.χ.:

Πρόγραμμα που στέλνει κείμενα στον εκτυπωτή

Πρόγραμμα που φορτώνει δεδομένα από τον Σκληρό Δίσκο στην Κύρια Μνήμη

Πρόγραμμα που εγκαθιστά την κάμερα (Drivers)

Πρόγραμμα που επιλέγει ποια διεργασία θα εκτελέσει η ΚΜΕ

Λογισμικό Εφαρμογών

π.χ.:

Πρόγραμμα Επεξεργασίας Κειμένων (word processor)

Ιστοπλοηγός (web browser)

Πρόγραμμα ηλεκτρονικού ταχυδρομείου

Πρόγραμμα ζωγραφικής

Πρόγραμμα διαχείρισης βάσης δεδομένων

Πρόγραμμα επεξεργασίας φύλλων εργασίας (Excel)

Λογισμικό: Δύο Κατηγορίες

6

Λογισμικό Συστημάτων

Είναι τα προγράμματα που εξυπηρετούν τις ανάγκες του Υλικού του Η/Υ και παρέχουν τις αναγκαίες υπηρεσίες που χρειάζεται το **Λογισμικό Εφαρμογών** από το Υλικό

Η ύπαρξη και λειτουργία του **δεν γίνεται άμεσα αντιληπτή στον χρήστη.**

Χρησιμοποιείται κατά την κατασκευή ή εκτέλεση του **Λογισμικού Εφαρμογών.**

Λογισμικό Εφαρμογών

Είναι τα προγράμματα που εξυπηρετούν **ανάγκες του γενικού χρήστη.**

Ο χρήστης **έρχεται σε επαφή μαζί του καθημερινά.**

Κατά την εκτέλεσή ή κατασκευή του **χρησιμοποιεί το Λογισμικό Συστημάτων.**

Λογισμικό: Δύο Κατηγορίες

7

Λογισμικό Συστημάτων

Λειτουργικά Συστήματα (ΛΣ, *Operating Systems*): Unix, Windows, MacOS, κ.ά.

Οδηγοί Υλικού (*Device Drivers*): Μεσολαβούν μεταξύ ΛΣ και μιας συσκευής.

Προγράμματα Υπηρεσιών (*utilities*): π.χ. Μορφοποίηση δίσκου, παύση διεργασιών.

Μεταγλωττιστές, Μεταφραστές (*assemblers, compilers*): π.χ. μεταφραστής C++.

Λογισμικό Εφαρμογών

Λογισμικό Γενικής Χρήσης: π.χ. Emacs, Acrobat Reader, Firefox, Chrome, κ.ά.

Εξειδικευμένο Λογισμικό: π.χ. Διαχείριση εστιατορίου, διαχείριση αποθήκης, λογιστικά πακέτα, κ.ά.

Ολοκληρωμένο Λογισμικό: Συλλογή εφαρμογών με συγγενές αντικείμενο, π.χ. MS Office, Lotus Notes.

Λειτουργικά Συστήματα

8

Το Λειτουργικό σύστημα (ΛΣ, *Operating System*, OS):

- Είναι ο **Πυρήνας του Λογισμικού Συστήματος** και προσφέρει το **Γραφικό Περιβάλλον** μέσω του οποίου ο χρήστης επικοινωνεί με τον Η/Υ.
- Είναι ένα **σύνολο προγραμμάτων** που:
 - Προσφέρουν το περιβάλλον μέσω του οποίου το **Λογισμικό Εφαρμογών** επικοινωνεί με το **Υλικό του Η/Υ**.
 - **Ελέγχουν και επιβλέπουν το Υλικό του Η/Υ** παρέχοντας διάφορες Υπηρεσίες (π.χ., ρύθμιση των παραμέτρων του υλικού του Η/Υ ώστε να λειτουργεί αποδοτικά).
 - **Διευθύνουν, Διαχειρίζονται και κατανέμουν τους πόρους του Η/Υ**, όπως η Μνήμη, οι Συσκευές Εισόδου και Εξόδου και η Κεντρική Μονάδα Επεξεργασίας, στα προγράμματα που ζητούν να τους χρησιμοποιήσουν.

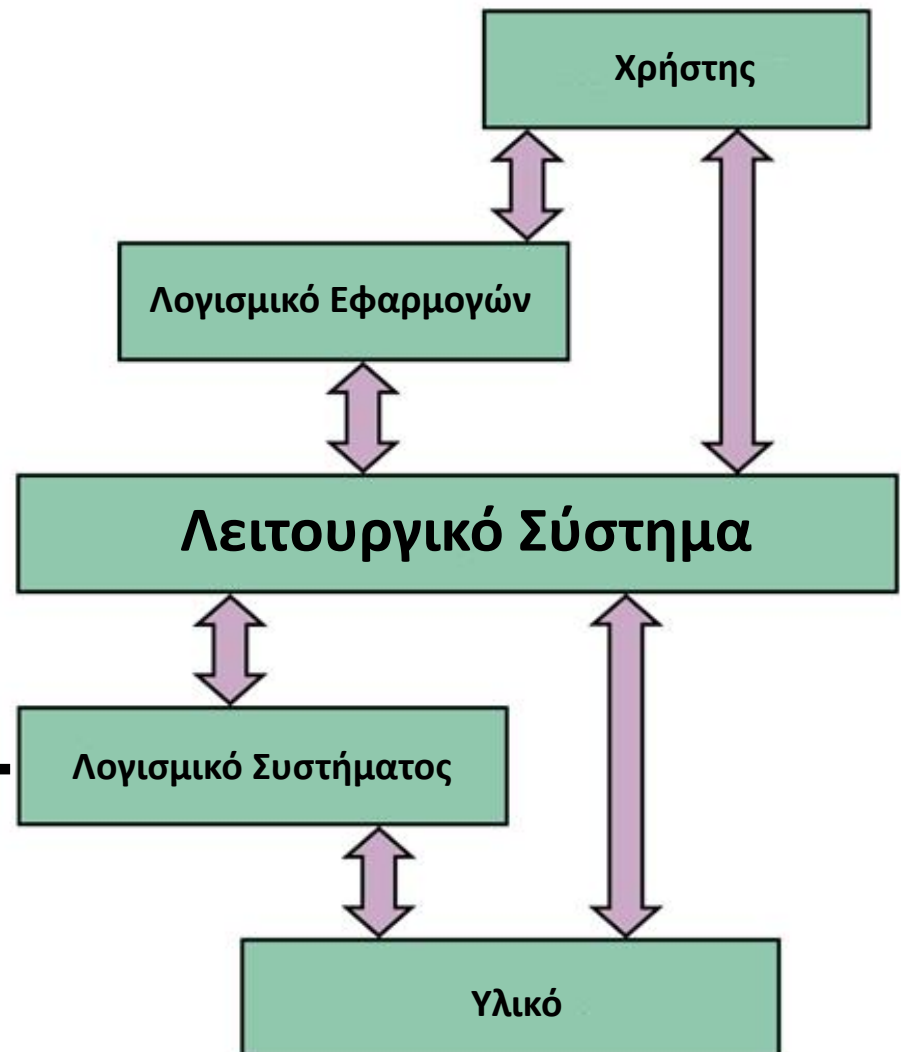
Η κεντρική ιδέα που αποτελεί τη **βάση λειτουργίας του Λειτουργικού Συστήματος** είναι ο “**Δίκαιος Καταμερισμός**” πόρων

Λειτουργικά Συστήματα

9

Το Λειτουργικό Σύστημα
Επικοινωνεί και
Αλληλεπιδρά με
διάφορα μέρη του Η/Υ.

π.χ. Βιβλιοθήκη
γραφικών, Οδηγοί
(Drivers) Συσκευών



Λειτουργικά Συστήματα

10

Γραφικό
Περιβάλλον

ΛΣ

Λογισμικό
Εφαρμογών

Ηλεκτρονικό
Ταχυδρομείο

Διαχείριση Βάσης
Δεδομένων

Πρόγραμμα
Ζωγραφικής

Επεξεργαστής
Κειμένου

Πλοηγός
Ιστού

Πυρήνας Λογισμικού Συστήματος

Λογισμικό
Συστημάτων

Οδηγός

Οδηγός

Οδηγός

Οδηγός

Οδηγός

Οδηγός

Οδηγός

Οδηγός

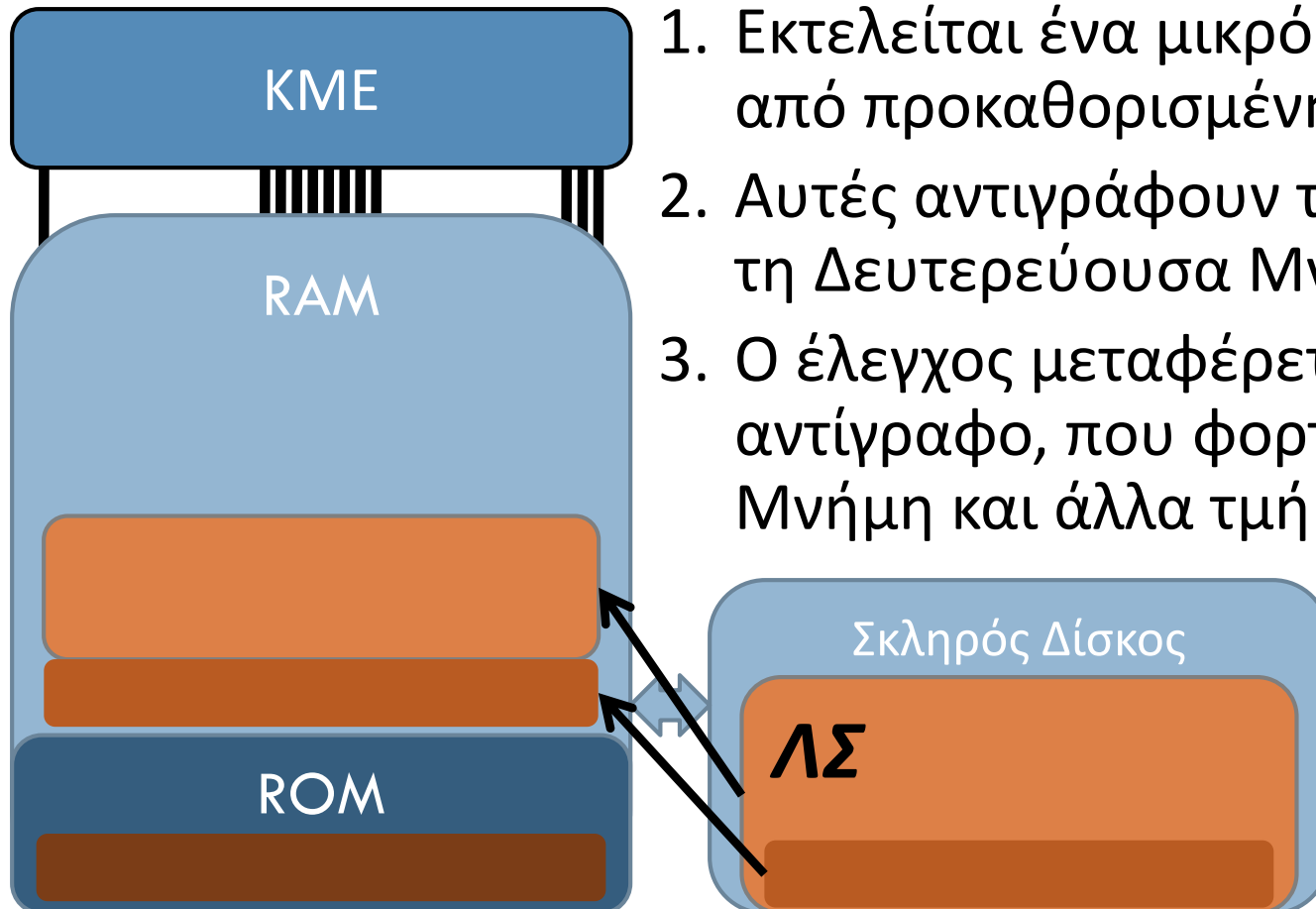


Εκκίνηση Λειτουργικού Συστήματος

11

Το ΛΣ αναλαμβάνει τον έλεγχο λίγο μετά την εκκίνηση του Η/Υ. Αναλυτικά, κατά την εκκίνηση (booting process):

1. Εκτελείται ένα μικρό σύνολο εντολών από προκαθορισμένη θέση της ROM.
2. Αυτές αντιγράφουν τμήμα του ΛΣ από τη Δευτερεύουσα Μνήμη στην Κύρια.
3. Ο έλεγχος μεταφέρεται σε αυτό το αντίγραφο, που φορτώνει στην Κύρια Μνήμη και άλλα τμήματα του ΛΣ.

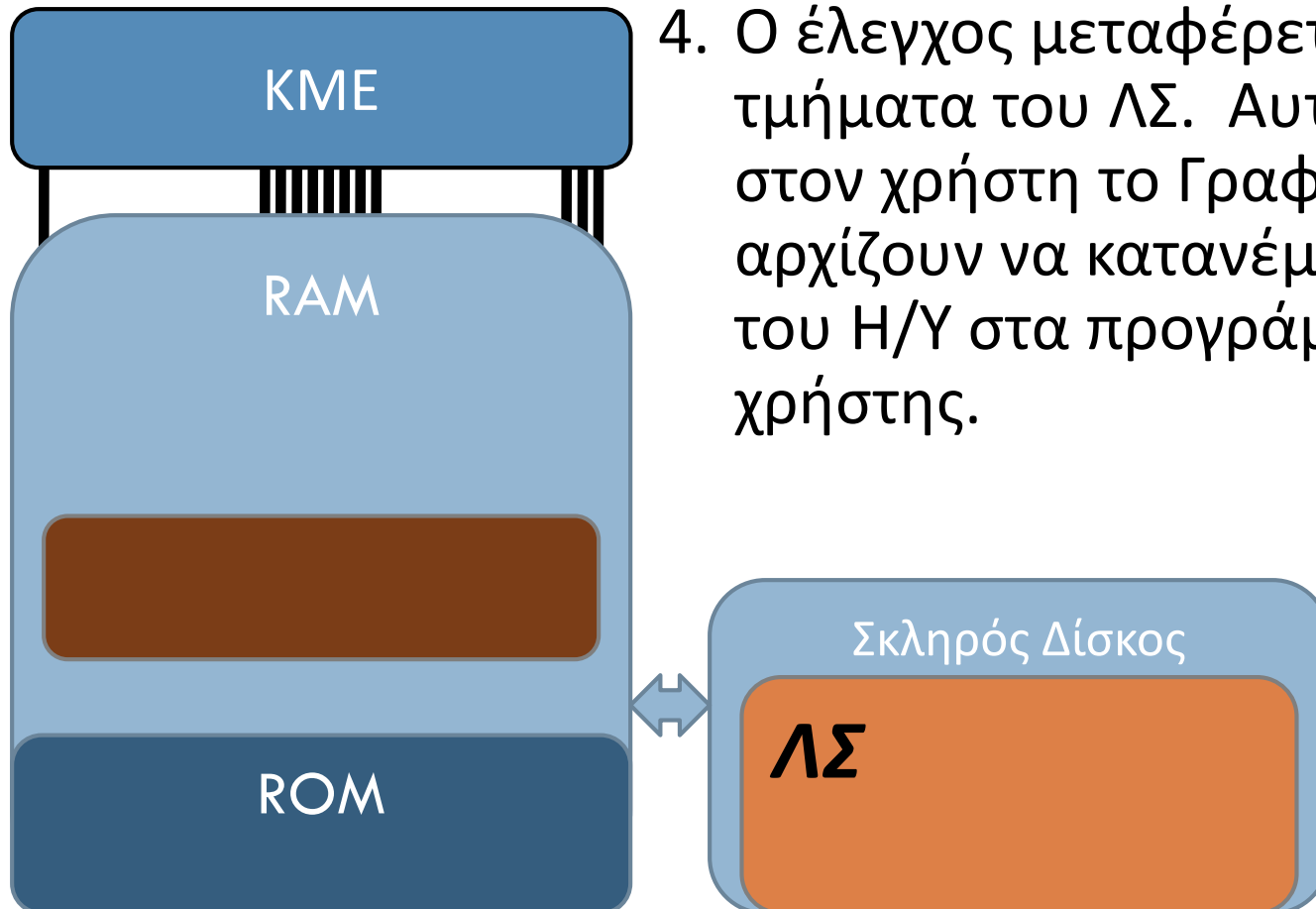


Εκκίνηση Λειτουργικού Συστήματος

12

Το ΛΣ αναλαμβάνει τον έλεγχο λίγο μετά την εκκίνηση του Η/Υ. Αναλυτικά, κατά την εκκίνηση (booting process):

4. Ο έλεγχος μεταφέρεται στα φορτωμένα τμήματα του ΛΣ. Αυτά παρουσιάζουν στον χρήστη το Γραφικό Περιβάλλον και αρχίζουν να κατανέμουν τους πόρους του Η/Υ στα προγράμματα που καλεί ο χρήστης.

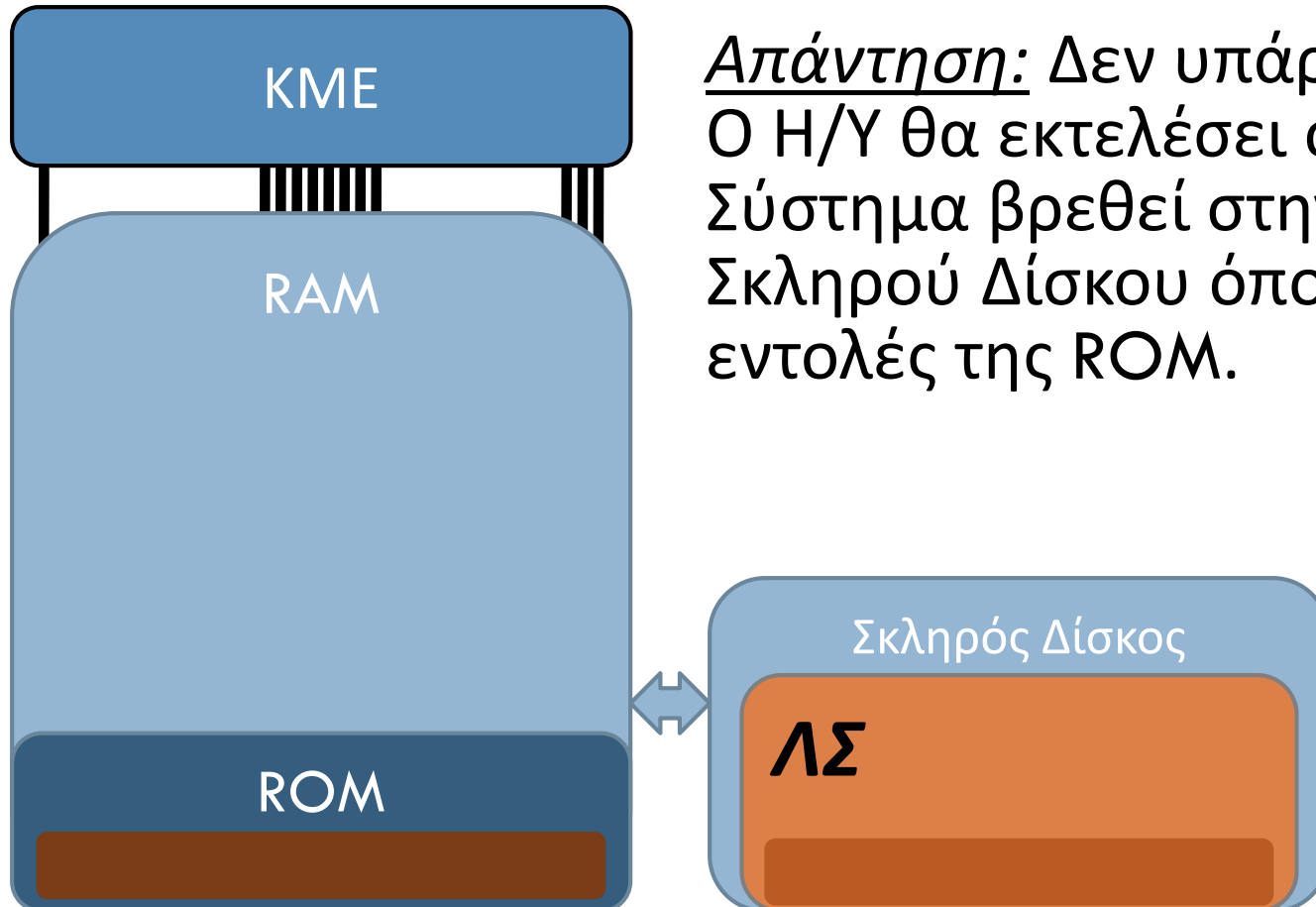


Εκκίνηση Λειτουργικού Συστήματος

13

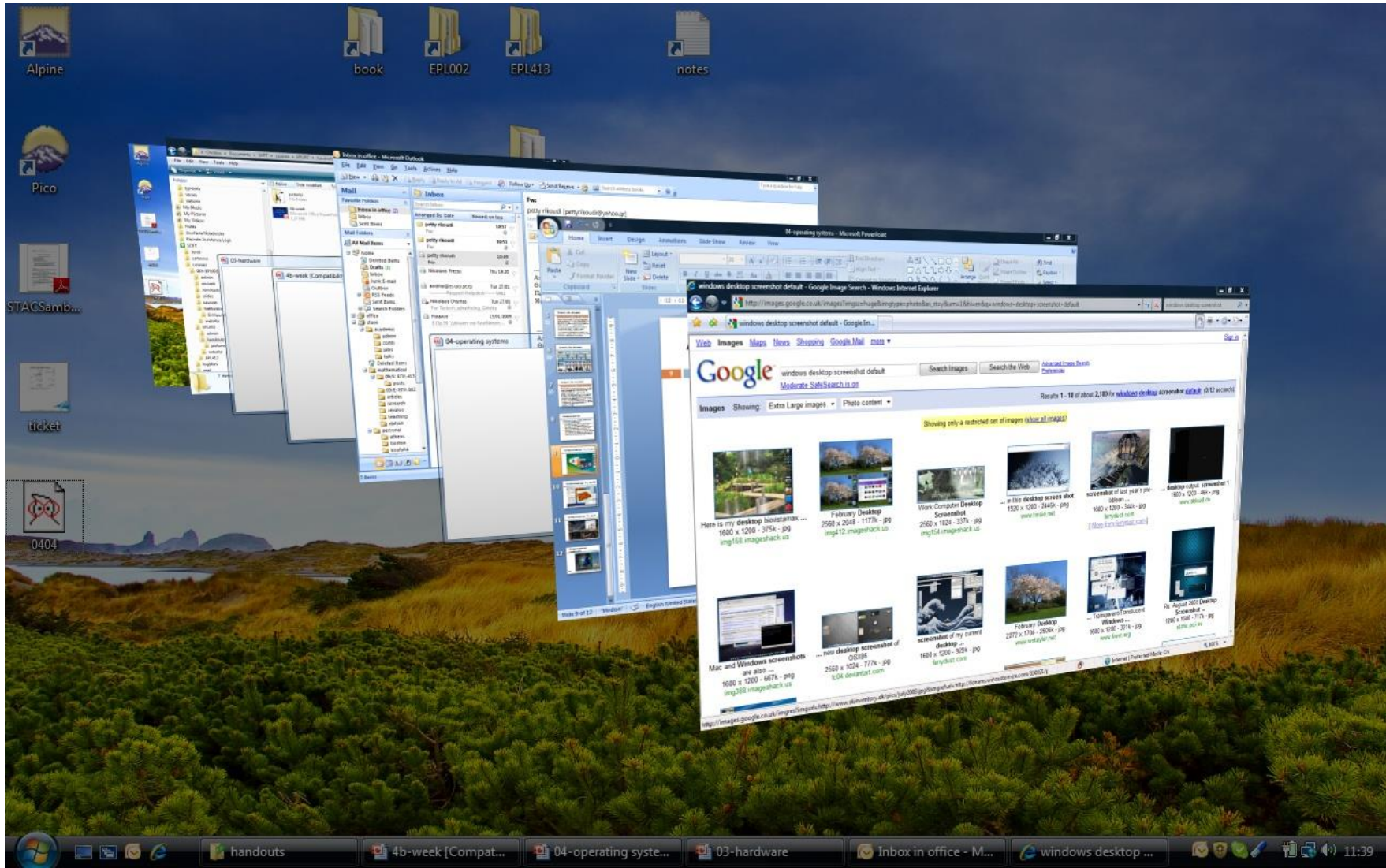
Ερώτηση: Ένας Η/Υ μπορεί να εκτελεί μόνο κάποιο συγκεκριμένο Λειτουργικό Σύστημα;

Απάντηση: Δεν υπάρχει περιορισμός. Ο Η/Υ θα εκτελέσει όποιο Λειτουργικό Σύστημα βρεθεί στην περιοχή του Σκληρού Δίσκου όπου ψάχνουν οι εντολές της ROM.



Παραδείγματα ΛΣ: Windows

14



Παραδείγματα ΛΣ: Mac OS

15

The image shows a Mac OS X desktop environment with several windows open. The central focus is a 3D visualization of atomic force microscopy (AFM) data, showing a textured surface with a grid-like pattern of peaks and valleys, rendered in shades of orange and yellow. This visualization is displayed in a window titled 'MesaGL: Ch2: /Volumes/Atomic_1/SIO-Si100/20021231SOI500/SOI500-1A008Topo-p.nc Q4 Short[2]'. To the left, the 'Gxsm - STM/AFM/SPA-LEED...' window is open, showing various scan parameters such as Range XY (153.2 nm), Steps XY (0.08 nm), Points XY (2000), and VRange Z (20 A). Below this, the 'GL Scene Setup' window is visible, with tabs for View, Light, Surface Material, Box Material, and Render Opt. On the right side, the 'Channel Selector' window is open, displaying a table of channels and their settings. Below that, the 'About This Mac' window is open, showing the Mac OS X Version 10.3.2 logo and system specifications: Processor Dual 2 GHz PowerPC G5 and Memory 2.5 GB DDR SDRAM. At the bottom right, a Finder window titled 'Macintosh HD' is open, showing a list of files and folders with columns for Name, Date Modified, and Size. The desktop background is blue, and the dock at the bottom contains various application icons. The system status bar at the top right shows the time as Tue 10:28 AM.

Ch.	View	Mode	Dir
1	Grey 2D	On	->
2	Surface 3D	Active	2>
3	Grey 2D	Off	->
4	Grey 2D	Off	->
5	Grey 2D	Off	->
6	Grey 2D	Off	->
7	Grey 2D	Off	->
8	Grey 2D	Off	->
9	Grey 2D	Off	->
10	Grey 2D	Off	->
11	Grey 2D	Off	->
12	Grey 2D	Off	->

Property	Value
Processor	Dual 2 GHz PowerPC G5
Memory	2.5 GB DDR SDRAM

Name	Date Modified	Size
Applications	Feb 26, 2004, 2:56 PM	
Applications (Mac OS 9)	Sep 26, 2003, 9:02 PM	
Developer	Feb 19, 2004, 9:23 AM	
Library	Dec 17, 2003, 3:00 PM	
sw	Feb 27, 2004, 5:00 PM	
System	Feb 15, 2004, 12:16 AM	
System Folder	Feb 17, 2004, 3:42 PM	
User Guides And Information	Oct 3, 2003, 11:15 PM	4 K
Users	Feb 18, 2004, 4:31 PM	

Παραδείγματα ΛΣ: *Ubuntu*

16

The screenshot displays the Ubuntu 8.10 desktop environment. The background is a scenic image of a harbor at dusk. In the top-left corner, there are icons for Applications, Places, and System. The desktop contains several icons: a 318.0 GB Media icon, a folder icon labeled '000300_m9_native_packed-1_en-US_9358', a 'Nou-090208.tar.bz2' icon, a 'photo_ren.sh' icon, and a 'wordpress.2008-10-13.xml' icon. A 'Videos - File Browser' window is open, showing a list of files and folders in the 'Videos' directory, including 'aeon_flux_d1', 'aeon_flux_d2', 'aeon_flux_d3', 'futurama', '310-to-yuma.avi', '310-to-yuma.ogg', 'Angus and Sadie.mov', 'baron_munchausen.avi', and 'buckaroo-02.ogg'. A terminal window titled 'jim@intrepid: ~' is open, displaying the output of the './info.sh' script. The system status bar at the bottom shows system information such as 'Sustent: 2.6.27-7-generic', 'Load: 0.34 0.27 0.36', 'CPU Speed: 1000mhz', 'CPU Usage: 17%', 'Core 0: 22C', 'Core 1: 22C', 'Root: 14,25GiB Free', 'Home: 55,18GiB Free', 'Down: 2 k/s Total: 779,19MiB', and 'Up: 0 k/s Total: 26,52MiB'.

```
./info.sh
LINUX
  Distro:      Ubuntu 8.10 intrepid
  Kernel:     2.6.27-7-generic
  Uptime:     days and 22:38:10 hrs

HARDWARE
  CPU:        AMD Athlon(tm) 64 X2 Dual Core Processor 4800+
  GPU:        GeForce 7600 GS
  RAM:        2088 mb
  Bogomips:  2006.04 bogomips

DESIGN
  Environment: Gnome
  GTK2:        Glider
  Icons:       Nou
  Application Font: Sans 10
  Terminal Font: Monospace 12
  Wallpaper:  the port flipped2560x1600
  Resolution:  1920x1200 px

Sustent: 2.6.27-7-generic
Uptime:  3h 47m 39s

Load: 0.34 0.27 0.36
RPM: 39 % Swap: 0 %

CPU Speed: 1000mhz
CPU Usage: 17 %

Core 0: 22C
Core 1: 22C

Root: 14,25GiB Free
Home: 55,18GiB Free

Down: 2 k/s Total: 779,19MiB
Up: 0 k/s Total: 26,52MiB
```

Παραδείγματα ΛΣ: Linux

17

The image shows a Linux desktop environment with a green and yellow background. A large analog clock is in the top left. A terminal window titled 'prophecy : bash' displays the output of the 'ls -l' command:

```
prophecy@localhost ~ $ ls -l
total 196
drwxr-xr-x  8 prophecy prophecy 4096 Jul 30 09:35 Check
drwxr-xr-x  6 prophecy prophecy 4096 Jul 31 09:15 Desktop
drwxr-xr-x  2 prophecy prophecy 4096 Jul 31 09:32 Downloads
lrwxrwxrwx  1 prophecy prophecy   29 Jul 27 21:49 FNAL -> /work/cartman-clued0/prophecy
drwxr-xr-x 25 prophecy prophecy 4096 Jul 27 04:01 MN_FIT
drwxr-xr-x  4 prophecy prophecy 4096 Jul 29 12:25 Zdetermination
drwxr-xr-x  7 prophecy prophecy 4096 Jul 29 18:10 bin
-rw-r--r--  1 prophecy prophecy 2357 Jul 29 23:57 higz_windows.dat
-rw-r--r--  1 prophecy prophecy 1036 Jul 29 10:36 src
-rw-r--r--  1 prophecy prophecy 1128 Jul 29 11:28 try.pdf
```

A Mozilla Firefox browser window shows the Pandora Radio website. The page title is 'Pandora Radio - Listen to Free Internet Radio; Find New Music'. The URL is 'http://www.pandora.com/'. The page content includes a search bar, a 'Create a New Station...' button, and several music recommendations such as 'Lacrymosa', 'Gates Of Dawn', and 'I Am The Day'. A system menu is open in the bottom left, showing options like 'Web Browser', 'System Settings', 'File Manager', 'Mozilla Firefox (bin)', 'Terminal', and 'XChat IRC'. A file manager window is open on the right, showing a directory structure with folders like 'Analysis', 'CAFe', 'Downloads', 'Collie', and 'CVS'. The system tray at the bottom shows the time as 15:03 and the active window as 'Pandora Radio - Listen to Free Internet Ra...'. The terminal window at the bottom right shows 'prophecy : bash'.

Παραδείγματα ΛΣ: *Palm OS*

18



Παραδείγματα ΛΣ: Symbian OS

19



Παραδείγματα ΛΣ: Apple iOS

20

iOS 4



Παραδείγματα ΛΣ: *Android OS*

21



Παραδείγματα ΛΣ: Mobile Windows

22

 Windows 8
Phone



Κατηγορίες ΛΣ

23

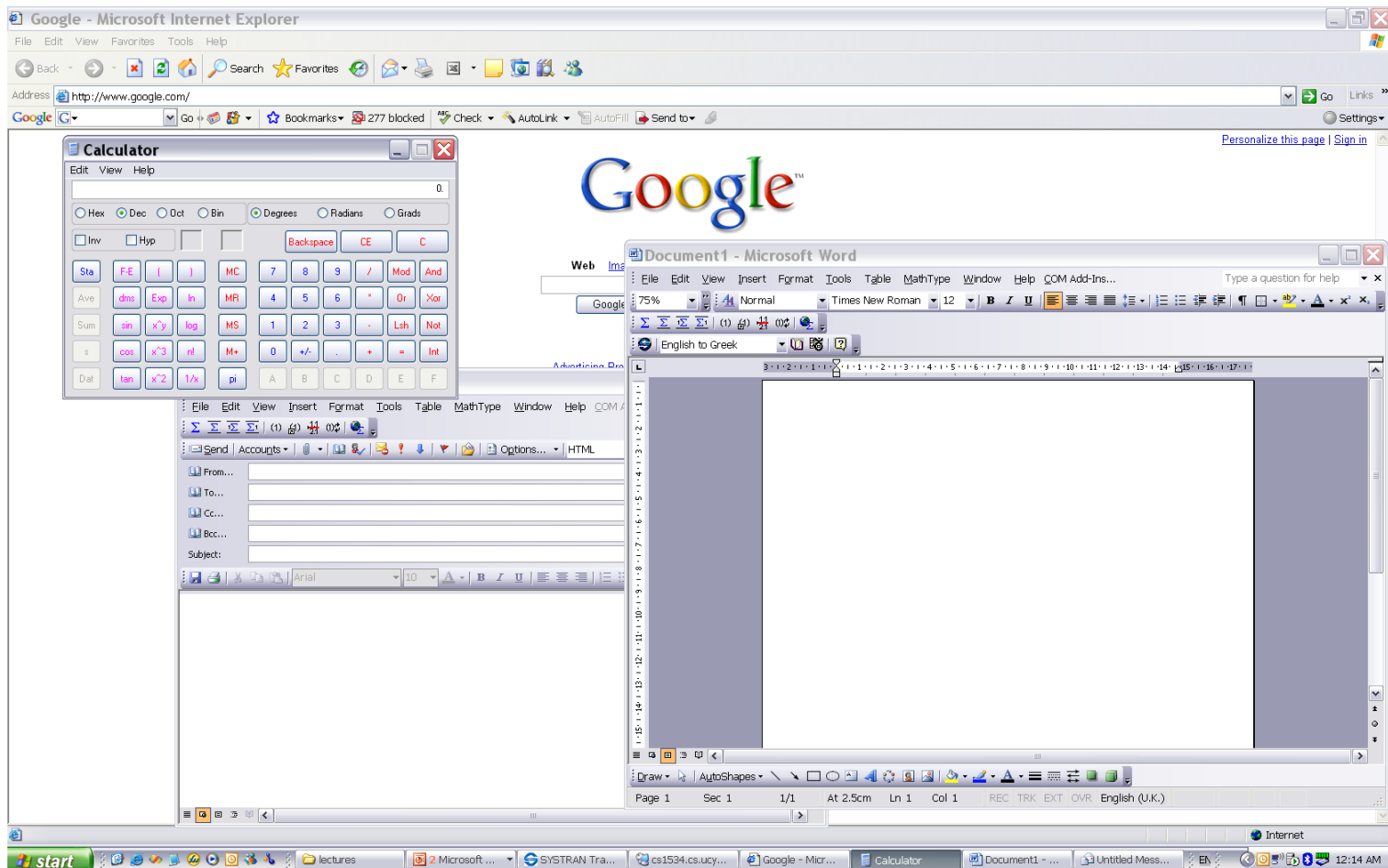
Τα Λειτουργικά Συστήματα μπορούν να διακριθούν:

- Με βάση το **Πλήθος Των Χρηστών** που μπορούν να χρησιμοποιήσουν τον Η/Υ, σε:
 1. Ενός χρήστη (Single-User): Μόνο ένας χρήστης μπορεί να χρησιμοποιεί τον υπολογιστή σε μια δεδομένη χρονική στιγμή. Π.χ., PalmOS, SymbianOS, Android, iOS, Mobile Windows.
 2. Πολλών χρηστών (Multi-User): Ταυτόχρονη χρήση του υπολογιστή από πολλούς χρήστες. Π.χ., Windows, Ubuntu, Linux, MacOS.
- Με βάση το **Πλήθος Των Εργασιών** που μπορούν να εκτελούνται ταυτόχρονα:
 1. Μίας εργασίας (Single-Tasking): PalmOS, SymbianOS, Android, iOS, Mobile Windows.
 2. Πολλών εργασιών (Multi-Tasking): Windows, Ubuntu, Linux, MacOS.

Κατηγορίες ΛΣ

24

Παράδειγμα Λειτουργικού Συστήματος Πολλών Εργασιών (Windows)



Ερωτήσεις;

Εξέλιξη ΛΣ

26

- Ένα πρόγραμμα είναι ένα στατικό σύνολο από εντολές.
- Ένα πρόγραμμα που εκτελείται, βρίσκεται στην Κύρια Μνήμη και οι εντολές του εκτελούνται η μια μετά την άλλη με τον Κύκλο Φέρε-Εκτέλεσε.
- Διεργασία (process) είναι ένα πρόγραμμα στη φάση εκτέλεσης του.
- Πολυπρογραμματισμός είναι η ταυτόχρονη συνύπαρξη πολλών προγραμμάτων στην Κύρια Μνήμη τα οποία επιζητούν πρόσβαση (ανταγωνιζόμενα μεταξύ τους) στην ΚΜΕ ώστε να ‘εκτελεστούν’
- Διαχείριση μνήμης είναι μια διαδικασία που παρακολουθεί ποια προγράμματα βρίσκονται στη Κύρια Μνήμη καθώς και την ακριβή θέση τους σε αυτή.

Ένα μοντέρνο Λειτουργικό Σύστημα υιοθετεί τον πολυπρογραμματισμό και γι’ αυτό εκτελεί Διαχείριση Μνήμης.

Εξέλιξη ΛΣ

27

- Τα Λειτουργικά Συστήματα έφθασαν να έχουν την σημερινή τους πολυπλοκότητα και ποικιλία μετά από δεκαετίες εξέλιξης.
- Ας δούμε κάποια σημαντικά στάδια της εξέλιξης αυτής.

Εξέλιξη ΛΣ: Επεξεργασία κατά Δεσμίδες

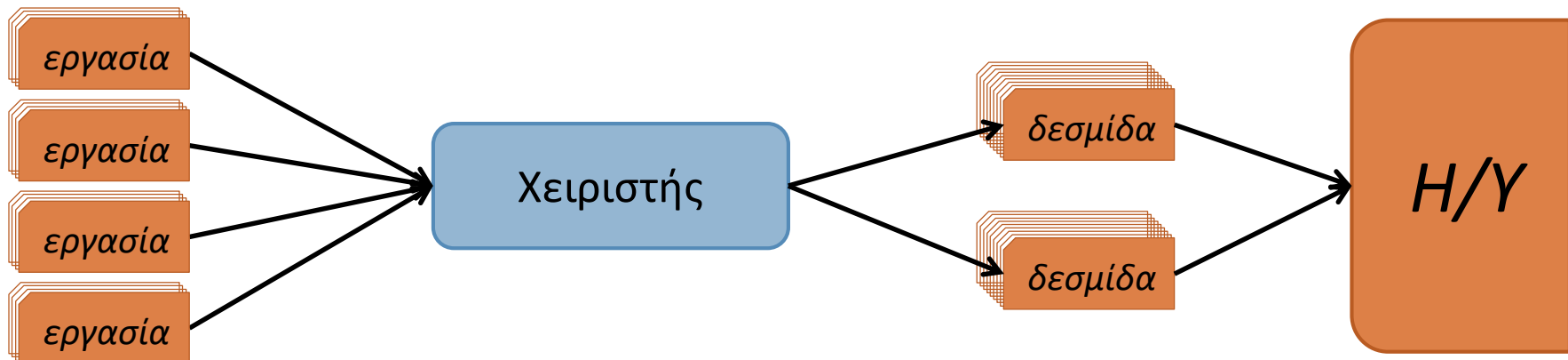
28

- Στις δεκαετίες '60-70, οι Η/Υ ήταν ογκώδεις, δύσχρηστοι, ακριβοί, και δυσεύρετοι και είχαν **ένα και μόνο ένα Χειριστή**.
- Ο Χειριστής θα έπαιρνε τις **διάφορες εργασίες από τους Χρήστες** και θα τις **οργάνωνε σε δεσμίδες (batches)**
- Μια **δεσμίδα** περιέχει **ένα σύνολο εργασιών που χρειάζονται τους ίδιους πόρους**.
 - Μια **εργασία** (job) αποτελείται από το πρόγραμμα, ένα σύνολο εντολών σχετικά με το Λογισμικό Συστήματος και τους πόρους που χρειαζόταν το πρόγραμμα για την εκτέλεση του.
- Οι δεσμίδες αυτές θα δίνονταν μετά στον Η/Υ για επεξεργασία.

Εξέλιξη ΛΣ: Επεξεργασία κατά Δεσμίδες

29

- ❑ Ο χειριστής δεν εκτελούσε κάθε εργασία χωριστά, αλλά τις ομαδοποιούσε σε δεσμίδες (batches) ανάλογα με τους πόρους που απαιτούσαν, και εκτελούσε τις δεσμίδες.
- ❑ Η οργάνωση σε δεσμίδες γινόταν για να μη χρειάζεται ο χειριστής να αλλάζει την κατάσταση της μηχανής πριν κάθε εκτέλεση.
- ❑ Έτσι, αντί να ετοιμάζει τον Η/Υ πριν την εκτέλεση κάθε εργασίας, τον ετοίμαζε πριν την εκτέλεση κάθε δεσμίδας.



Εξέλιξη ΛΣ: Επεξεργασία κατά Δεσμίδες

30



Ο ρόλος του ΛΣ ήταν απλώς να φορτώνει τις εργασίες της δεσμίδας στην Κύρια Μνήμη και να μεταφέρει τον έλεγχο (Μονοπρογραμματισμός):

1. Τον έλεγχο έχει το ΛΣ.
2. Το ΛΣ φορτώνει στη Κύρια Μνήμη την εργασία#1 και της δίνει τον έλεγχο.
3. Η εργασία#1 τερματίζει και ο έλεγχος επιστρέφει στο ΛΣ.
4. Το ΛΣ φορτώνει στη Κύρια Μνήμη την εργασία#2 και της δίνει τον έλεγχο.
5. Η εργασία#2 τερματίζει και ο έλεγχος επιστρέφει στο ΛΣ.
6. ...

Εξέλιξη ΛΣ: Πολυπρογραμματισμός

31



Πρόβλημα: Έστω ότι η εργασία#2 ζητάει από τον εκτυπωτή να τυπώσει έναν μεγάλο όγκο δεδομένων.

Όσο τα δεδομένα μεταφέρονται στον εκτυπωτή, η ΚΜΕ μένει ανενεργή --- παρόλο που η εργασία#3 περιμένει...

Λύση: **Πολυπρογραμματισμός** (*multi-programming*).

Οι εργασίες φορτώνονται ταυτόχρονα στην Κύρια Μνήμη. Όταν αυτή που έχει τον έλεγχο δεν χρησιμοποιεί την ΚΜΕ (επειδή περιμένει κάποια Ε/Ε), το ΛΣ μεταφέρει τον έλεγχο στην επόμενη.

Εξέλιξη ΛΣ: Πολυπρογραμματισμός

32



Π.χ.:

1. Τον έλεγχο έχει το ΛΣ και φορτώνει τις εργασίες (διεργασίες) στη Κύρια Μνήμη.
2. Το ΛΣ δίνει τον έλεγχο στη διεργασία#1.
3. Η διεργασία#1 αρχίζει να τυπώνει και ο έλεγχος επιστρέφει στο ΛΣ.
4. Το ΛΣ δίνει τον έλεγχο στη διεργασία#2.
5. Η διεργασία#2 τερματίζει και ο έλεγχος επιστρέφει στο ΛΣ.
6. ...

Ένα πρόγραμμα που έχει φορτωθεί στην Κύρια Μνήμη για να εκτελεστεί λέγεται διεργασία (process).

Εξέλιξη ΛΣ: Χρονομερισμός

33



Πρόβλημα: Έστω ότι η διεργασία#2 απασχολεί την ΚΜΕ διαρκώς για ώρες, ενώ η διεργασία#3 την χρειάζεται μόνο για λίγα λεπτά.

Η διαχείριση αδικεί την διεργασία#3...

Λύση: **Χρονομερισμός** (*time sharing*).

Το ΛΣ μεταφέρει τον έλεγχο στην κάθε διεργασία μόνο για κάποιο μικρό χρόνο. Όταν αυτός εξαντληθεί, το ΛΣ ανακτά τον έλεγχο και τον μεταφέρει στην επόμενη διεργασία, πάλι για μικρό χρόνο, κ.ο.κ.

Εξέλιξη ΛΣ: Χρονομερισμός

34



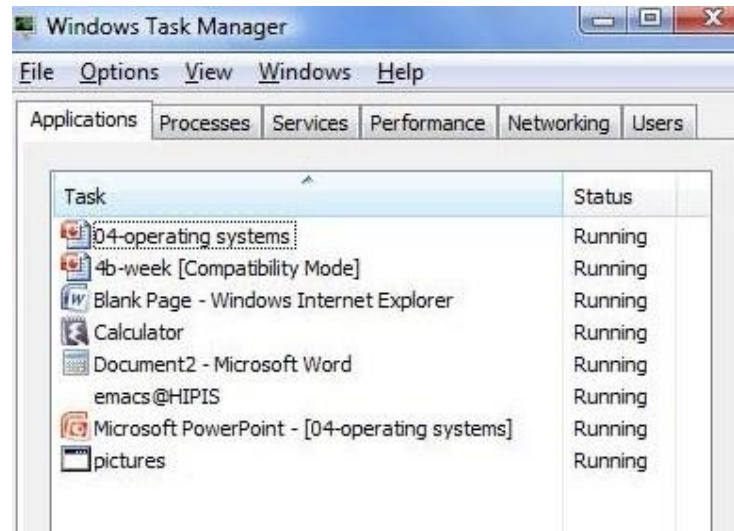
Η διαχείριση είναι πολύ πιο δίκαιη.

Αν η ΚΜΕ είναι αρκετά ισχυρή και διεργασίες σχετικά λίγες, η εναλλαγή μεταξύ τους δεν προκαλεί πρόβλημα. Ακόμη κι αν εκτελούν διαδραστικά προγράμματα, ο χρήστης βλέπει (στο τερματικό) σχεδόν ό,τι θα έβλεπε και αν η διεργασία του χρησιμοποιούσε την ΚΜΕ αποκλειστικά.

Αυτός ο νοητός αποκλειστικός Η/Υ που μοιάζει να έχει μπροστά του ο χρήστης λέγεται **νοητή μηχανή** (*virtual machine*).

Εξέλιξη ΛΣ: Διαχείριση Πόρων

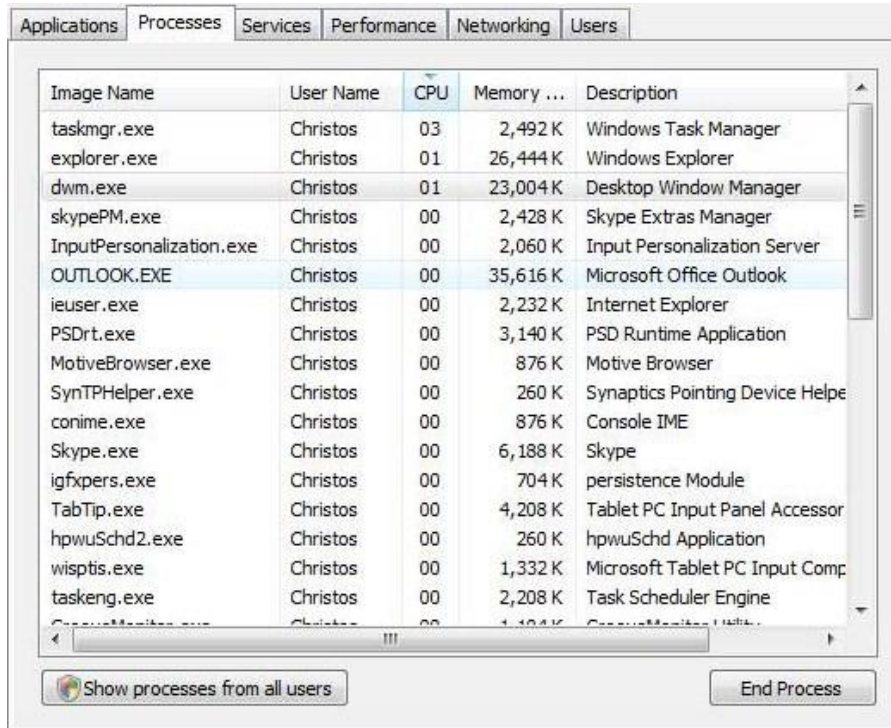
35



- Ως αποτέλεσμα της εξέλιξής τους, τα σύγχρονα ΛΣ έχουν φθάσει να επιτελούν ένα πολύ σύνθετο έργο:
 1. Παρακολούθησης όλων των διεργασιών που είναι φορτωμένες στην Κύρια Μνήμη, και
 2. Κατανομής των πόρων του συστήματος (ΚΜΕ, Κύρια Μνήμη, Δευτερεύουσα Μνήμη, Μονάδες Ε/Ε) στις διεργασίες αυτές.

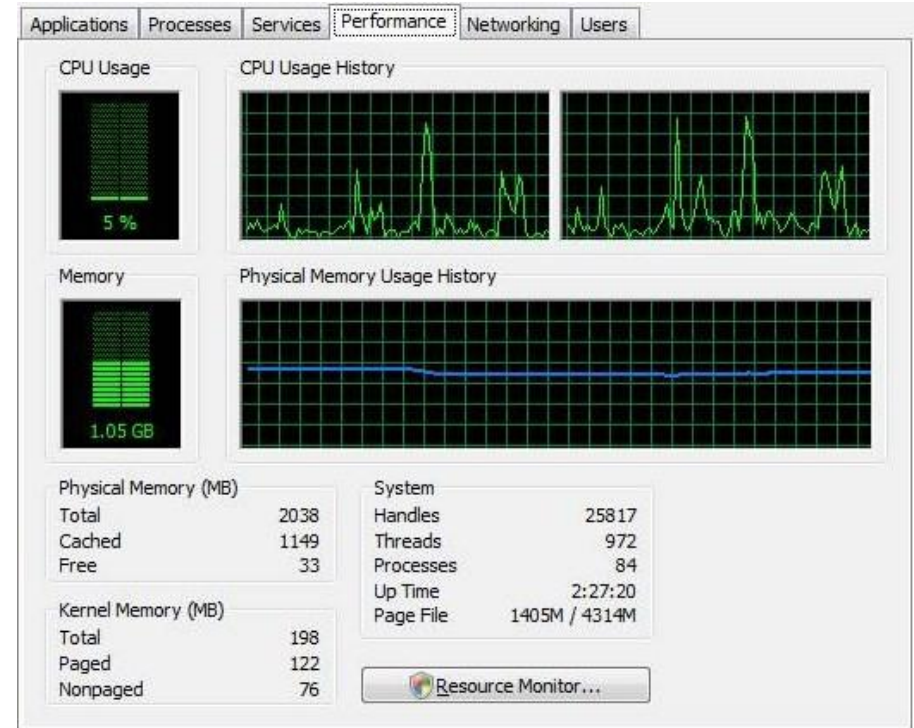
Εξέλιξη ΛΣ: Διαχείριση Πόρων

36



The screenshot shows the 'Processes' tab of the Windows Task Manager. It displays a list of running processes with columns for Image Name, User Name, CPU usage, Memory usage, and Description. The 'OUTLOOK.EXE' process is highlighted in blue.

Image Name	User Name	CPU	Memory ...	Description
taskmgr.exe	Christos	03	2,492 K	Windows Task Manager
explorer.exe	Christos	01	26,444 K	Windows Explorer
dwm.exe	Christos	01	23,004 K	Desktop Window Manager
skypePM.exe	Christos	00	2,428 K	Skype Extras Manager
InputPersonalization.exe	Christos	00	2,060 K	Input Personalization Server
OUTLOOK.EXE	Christos	00	35,616 K	Microsoft Office Outlook
ieuser.exe	Christos	00	2,232 K	Internet Explorer
PSDrt.exe	Christos	00	3,140 K	PSD Runtime Application
MotiveBrowser.exe	Christos	00	876 K	Motive Browser
SynTPHelper.exe	Christos	00	260 K	Synaptics Pointing Device Helper
conime.exe	Christos	00	876 K	Console IME
Skype.exe	Christos	00	6,188 K	Skype
igfxpers.exe	Christos	00	704 K	persistence Module
TabTip.exe	Christos	00	4,208 K	Tablet PC Input Panel Accessor
hpwuSchd2.exe	Christos	00	260 K	hpwuSchd Application
wisptis.exe	Christos	00	1,332 K	Microsoft Tablet PC Input Comp
taskeng.exe	Christos	00	2,208 K	Task Scheduler Engine



The screenshot shows the 'Performance' tab of the Windows Performance Monitor. It displays various system performance metrics and their history graphs.

CPU Usage: 5%

Memory: 1.05 GB

Physical Memory (MB)

Total	2038
Cached	1149
Free	33

Kernel Memory (MB)

Total	198
Paged	122
Nonpaged	76

System

Handles	25817
Threads	972
Processes	84
Up Time	2:27:20
Page File	1405M / 4314M

Διεργασίες και Επίδοση Συστήματος, όπως φαίνονται στον Task Manager του ΛΣ Windows.

Ερωτήσεις;

Διαχείριση Μνήμης: Φόρτωση

38

.....

LOAD 200 R1

LOAD 201 R2

.....

+14

-10

.....

...

70

71

...

200

201

...

ΛΣ

Κάθε στιγμή, στη **Κύρια Μνήμη (RAM)** **συνυπάρχουν το ΛΣ και ≥ 0 διεργασίες.**

Το ΛΣ για να φορτώσει ένα πρόγραμμα στη Κύρια Μνήμη, **αντιγράφει τις λέξεις (δηλ. τις εντολές και τα δεδομένα) του προγράμματος μία προς μία, από τη Δευτερεύουσα Μνήμη στην Κύρια Μνήμη.**

Η αντιγραφή γίνεται σε όποια **Συνεχή Περιοχή** της Κύριας Μνήμης είναι βολική.

Ερώτηση: Πώς ξέρει το πρόγραμμα πού στη Κύρια Μνήμη θα βρεθούν οι εντολές και τα δεδομένα του;

Π.χ., πώς ήξερε ότι η εντολή «LOAD 200 R1» θα κατέληγε στη θέση 70 και ότι ο αριθμός “+14” θα κατέληγε στη θέση 200 της Κύριας Μνήμης;

Διαχείριση Μνήμης: Φόρτωση

39

.....

LOAD 200 R1

LOAD 201 R2

.....

+14

-10

.....

ΛΣ

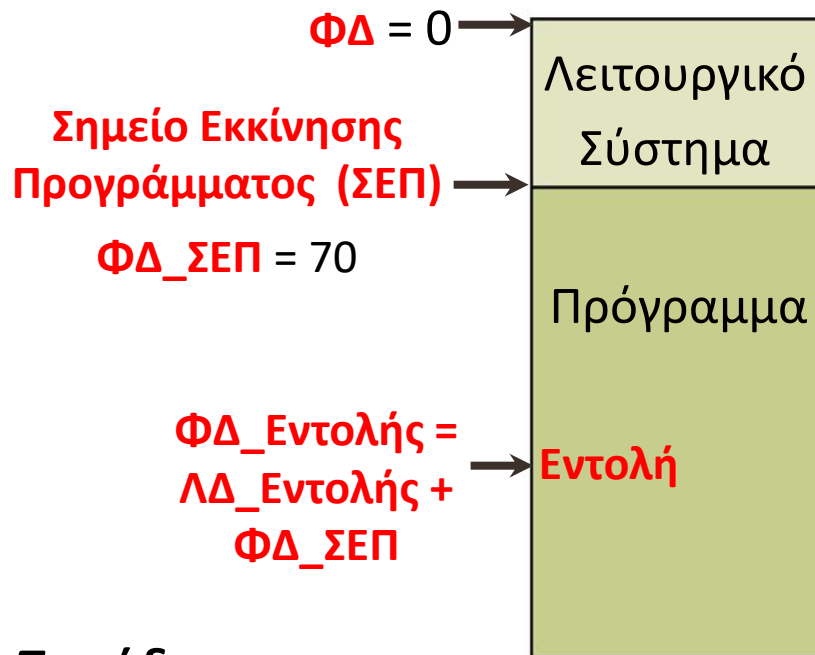
Απάντηση: Δεν ξέρει!

...
70 Στη Δευτερεύουσα Μνήμη, το πρόγραμμα αναφέρεται στις εντολές και τα δεδομένα του μέσω **Λογικών Διευθύνσεων**: π.χ.
71 **Μέσω των θέσεών που έχουν εντός του προγράμματος.**

...
200 **Κατά τη φόρτωση του προγράμματος στην**
201 **Κύρια Μνήμη, το ΛΣ μετατρέπει τις Λογικές**
... **Διευθύνσεις (των εντολών και το**
δεδομένων) σε Φυσικές Διευθύνσεις, με
βάση το Σημείο Εκκίνησης του
Προγράμματος (ΣΕΠ) στην Κύρια Μνήμη
Η μετατροπή αυτή λέγεται **Συσχέτιση**
Διευθύνσεων (Address Binding).

Διαχείριση Μνήμης: Φόρτωση

40



Για να προσδιορίσουμε τη **Φυσική Διεύθυνση** της εντολής (**ΦΔ_Εντολής**) στην Κύρια Μνήμη, απλά προσθέτουμε τη **Λογική Διεύθυνση** της εντολής (δηλ. τη θέση που έχει στο πρόγραμμα, **ΛΔ_Εντολής**) στη **Φυσική Διεύθυνση** του Σημείου Εκκίνησης του Προγράμματος (**ΦΔ_ΣΕΠ**) στην Κύρια Μνήμη.

Παράδειγμα:

Έστω **ΛΔ_Εντολής = 30** (δηλαδή βρίσκεται στη θέση 30 του προγράμματος πριν φορτωθεί στην Κύρια Μνήμη)

Φυσική Διεύθυνση ΣΕΠ (ΦΔ_ΣΕΠ) όταν φορτώθηκε στη Κύρια Μνήμη : 70

$$\mathbf{\Phi\Delta\ \text{Εντολής (στην Κύρια Μνήμη) = 30 + 70 = 100}}$$

$$\mathbf{\Phi\Delta\ \text{Εντολής} = \Lambda\Delta\ \text{Εντολής} + \Phi\Delta\ \Sigma\text{ΕΠ}}$$

Διαχείριση Μνήμης: Συνεχής

41

RAM

διεργασία #1

ΛΣ

Σε ένα μόνο-προγραμματιστικό ΛΣ, κάθε στιγμή η Κύρια Μνήμη (RAM) περιέχει μόνο το ΛΣ και μία διεργασία, η οποία ελέγχει όλη τη RAM που δεν χρησιμοποιεί το ΛΣ.

Αυτό λέγεται **Διαχείριση Συνεχούς Μνήμης** (single contiguous memory management).

Έχει όλα τα μειονεκτήματα του μόνο-προγραμματισμού (π.χ., μη αποδοτική χρήση των πόρων του Υπολογιστή, κτλ). Επιπλέον:

Ερώτηση: Τι γίνεται αν το πρόγραμμα δεν χωράει στη μνήμη;

Απάντηση: Δεν εκτελείται...

Άλλοι Μέθοδοι Διαχείριση Μνήμης

42

- Άλλοι τρόποι διαχείρισης μνήμης παρέχουν περισσότερη ευελιξία
- **Διαμέριση Μνήμης:** Η Κύρια Μνήμη είναι χωρισμένη σε κομμάτια σταθερού μεγέθους ή σε κομμάτια που το μέγεθός τους καθορίζεται από τις ανάγκες των προγραμμάτων.
- **Σελιδοποίηση Μνήμης:** Τα προγράμματα χωρίζονται σε σελίδες σταθερού μεγέθους που φυλάσσονται σε πλαίσια μνήμης όταν φορτώνονται στη Κύρια Μνήμη.

Διαχείριση Μνήμης: Διαμέριση

43

διεργασία #4

διεργασία #5

διεργασία #2

διεργασία #1

ΛΣ

Σε ένα πολύ-προγραμματιστικό ΛΣ, κάθε στιγμή η Κύρια Μνήμη περιέχει το ΛΣ και πολλές διεργασίες.

Μια μέθοδος για να διαχειριστούμε την Κύρια Μνήμη είναι **να την χωρίσουμε σε διαμερίσματα, καθένα από τα οποία να φιλοξενεί μία διεργασία.**

Τα διαμερίσματα μπορεί να είναι *ισομεγέθη ή ανισομεγέθη, σταθερά ή μεταβλητά.*

Αυτό λέγεται **Διαμέριση** (*partitioning*).

Διαχείριση Μνήμης: Διαμέριση

44

διεργασία #4

διεργασία #5

διεργασία #2

διεργασία #1

ΛΣ

Μειονεκτήματα:

Με σταθερά διαμερίσματα, είναι δύσκολο να επιλέξουμε σωστό μέγεθος

- ❑ **Μικρά διαμερίσματα** \Rightarrow κάποια προγράμματα ίσως να μη χωρέσουν.
- ❑ **Μεγάλα διαμερίσματα** \Rightarrow πολλά κενά.
- ❑ **Μεταβλητά διαμερίσματα** \Rightarrow αυξάνει η πολυπλοκότητα του ΛΣ και η χρονική επιβάρυνση που επιφέρει.

Επιπλέον:

Ερώτηση: Τι γίνεται αν ένα πρόγραμμα δεν χωράει στη μνήμη;

Απάντηση: Όπως πριν, δεν εκτελείται...

Διαχείριση Μνήμης: Σελιδοποίηση

45

διεργασία #4, σελ. #5

διεργασία #4, σελ. #4

διεργασία #3, σελ. #2

διεργασία #3, σελ. #1

διεργασία #4, σελ. #3

διεργασία #4, σελ. #2

διεργασία #4, σελ. #1

διεργασία #1

ΛΣ

Μια άλλη μέθοδος είναι να χωρίσουμε και τα προγράμματα σε τμήματα:

- ❑ Κάθε πρόγραμμα χωρίζεται σε σελίδες (pages).
- ❑ Η Κύρια Μνήμη χωρίζεται σε πλαίσια (frames), με μέγεθος αυτό των σελίδων.

Το ΛΣ φορτώνει κάθε σελίδα ενός προγράμματος σε ένα πλαίσιο. Τα πλαίσια που φιλοξενούν ένα πρόγραμμα δεν είναι απαραίτητως γειτονικά.

Αυτό λέγεται **σελιδοποίηση** (paging).

Διαχείριση Μνήμης: Σελιδοποίηση

46

διεργασία #4, σελ. #5

διεργασία #4, σελ. #4

διεργασία #3, σελ. #2

διεργασία #3, σελ. #1

διεργασία #4, σελ. #3

διεργασία #4, σελ. #2

διεργασία #4, σελ. #1

διεργασία #1

ΛΣ

Ερώτηση: Τι γίνεται αν ένα πρόγραμμα δεν χωράει στη μνήμη;

Απάντηση: Μπορεί να εκτελεστεί! Πώς;

Το ΛΣ **εκτελεί το πρόγραμμα ακόμη και χωρίς να είναι φορτωμένες όλες του οι σελίδες** στην Κύρια Μνήμη. Όταν η εκτέλεση απαιτήσει σελίδα που δεν είναι στη Κύρια Μνήμη, η σελίδα φορτώνεται (σε ένα κενό πλαίσιο ή σε πλαίσιο άλλης, που δεν χρησιμοποιείται) και η εκτέλεση συνεχίζει.

Το αποτέλεσμα λέγεται **Εικονική Μνήμη** (*Virtual Memory*): η Κύρια Μνήμη «μοιάζει» τόσο μεγάλη όσο η Δευτερεύουσα.

Διαχείριση Διεργασιών

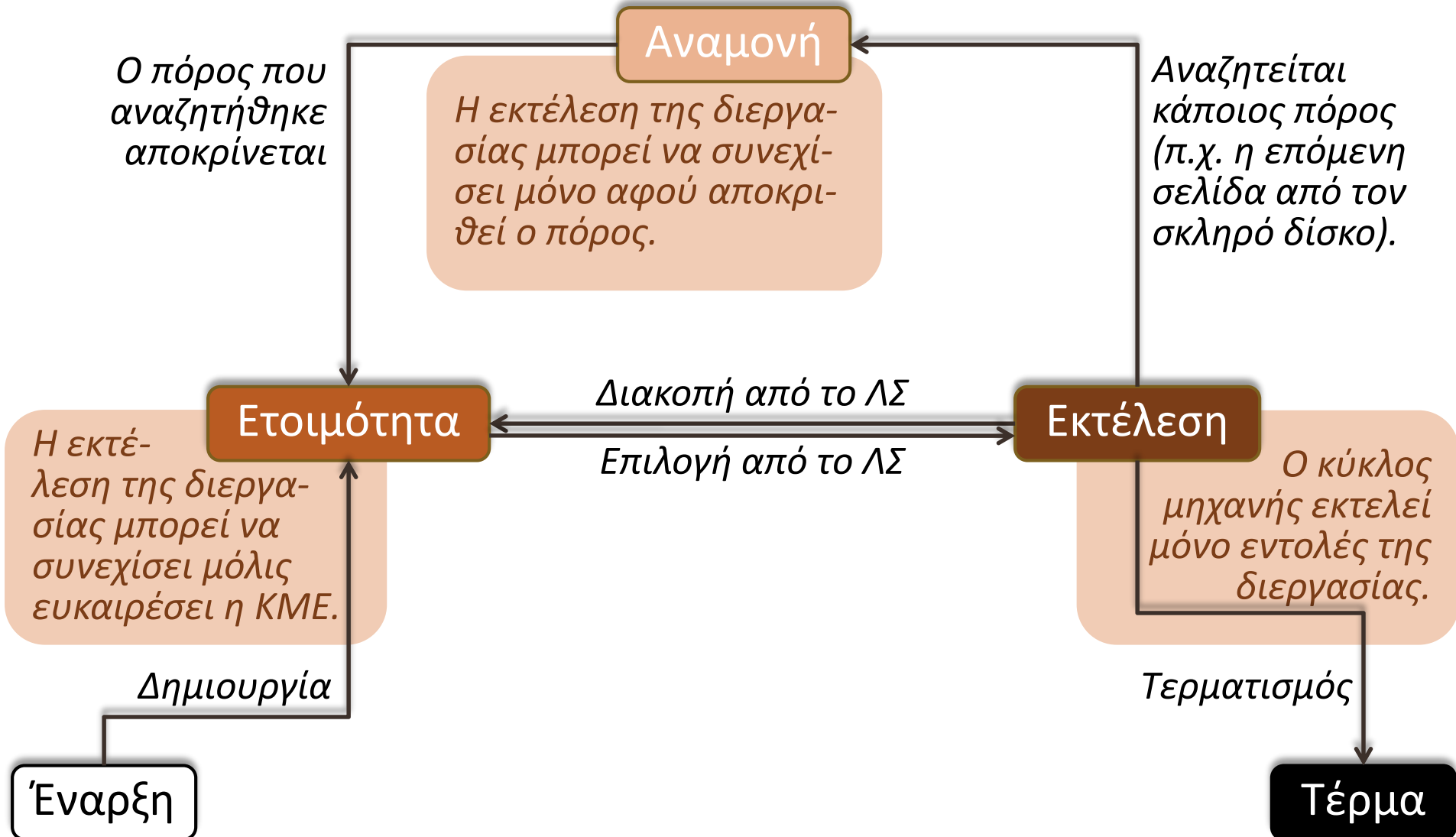
47

Ερώτηση: Πώς κατορθώνει το ΛΣ να παρακολουθεί την εκτέλεση πολλών διεργασιών στην ίδια ΚΜΕ;

Πριν απαντήσουμε, πρέπει να περιγράψουμε όλες τις διαφορετικές καταστάσεις στις οποίες μπορεί να βρεθεί μια διεργασία.

Διαχείριση Διεργασιών

48



Διαχείριση Διεργασιών

Για κάθε Κατάσταση (*Έναρξη, Ετοιμότητα, Αναμονή, Εκτέλεση, Τέρμα*), το ΛΣ διατηρεί και παρακολουθεί έναν Κατάλογο (Κατ_Έν, Κατ_Ετ, Κατ_Αν, Κατ_Εκτ, Κατ_Τέρ). Επίσης, για κάθε Διεργασία που βρίσκεται στη Κύρια Μνήμη, ανάλογα με την κατάσταση που βρίσκεται η Διεργασία, το ΛΣ διατηρεί για αυτήν **μια εγγραφή** στον αντίστοιχο κατάλογο.

Η εγγραφή περιέχει πληροφορίες για την Διεργασία, όπως π.χ.,:

- ▣ Πού βρίσκεται στη Κύρια Μνήμη,
- ▣ Ποια είναι η επόμενη προς εκτέλεση εντολή της,
- ▣ Τι περιείχαν οι καταχωρητές της ΚΜΕ την τελευταία φορά που διακόπηκε η εκτέλεση της διεργασίας.

Ερώτηση: Αν η Κύρια Μνήμη περιέχει **16** Διεργασίες και οι κατάλογοι για τις καταστάσεις *Έναρξη, Αναμονή, Τερματισμός* περιέχουν αντίστοιχα **4, 5,** και **1** διεργασίες, πόσες διεργασίες περιέχει ο κατάλογος για την *Ετοιμότητα*;

Διαχείριση Διεργασιών

50

16
διεργασίες
συνολικά

Αναμονή

5 διεργασίες

Ετοιμότητα

? διεργασίες

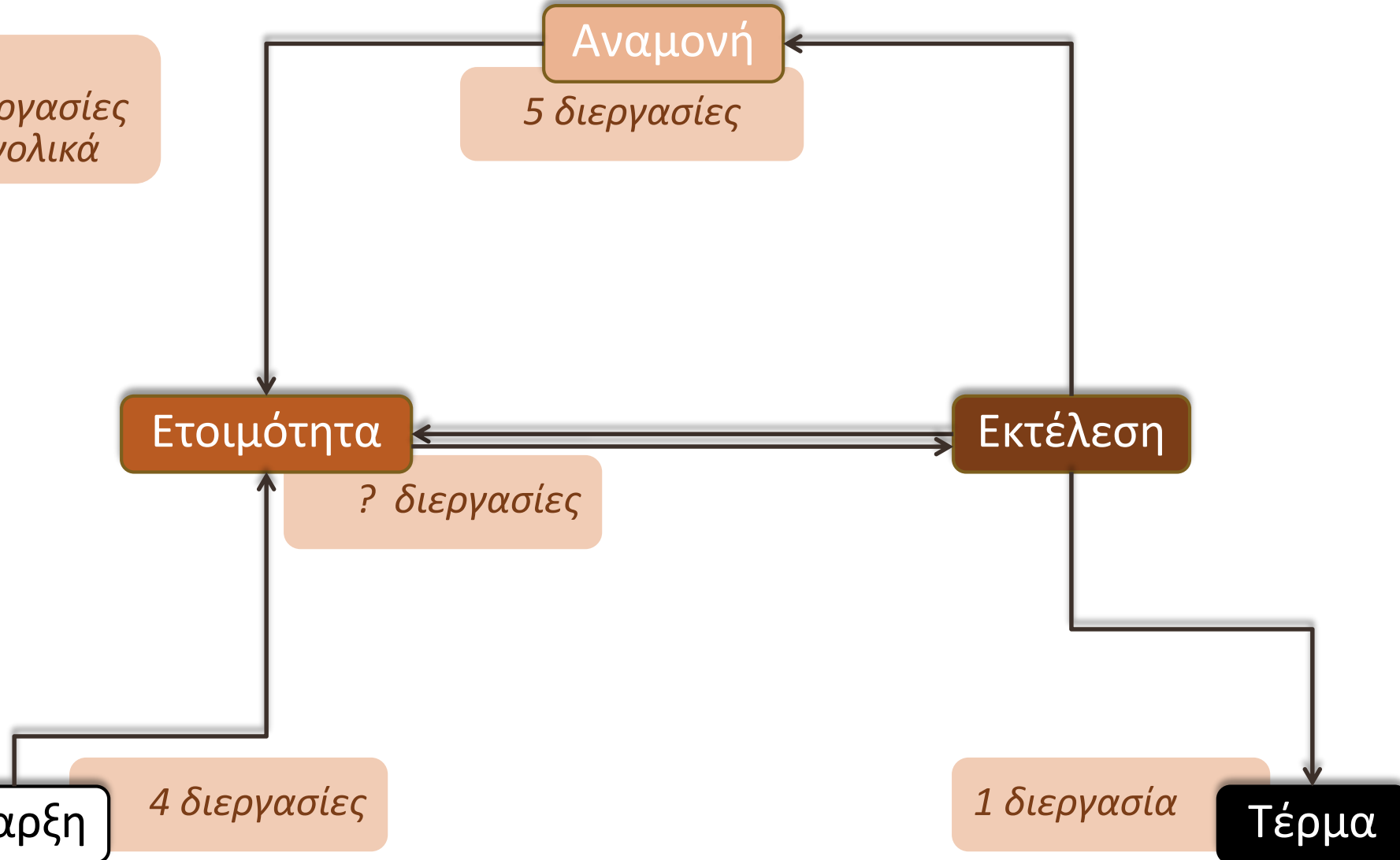
Εκτέλεση

Έναρξη

4 διεργασίες

1 διεργασία

Τέρμα



Διαχείριση Διεργασιών

51

Ερώτηση: Αν η Κύρια Μνήμη περιέχει **16** διεργασίες και οι κατάλογοι για τις καταστάσεις Έναρξη, Αναμονή, Τερματισμός περιέχουν αντίστοιχα **4, 5, και 1** διεργασίες, πόσες διεργασίες περιέχει ο κατάλογος για την *Ετοιμότητα*;

Απάντηση: 4 διεργασίες σε Έναρξη
5 διεργασίες σε Αναμονή
1 διεργασίες σε Τερματισμό
1 διεργασία σε Εκτέλεση (προφανώς)

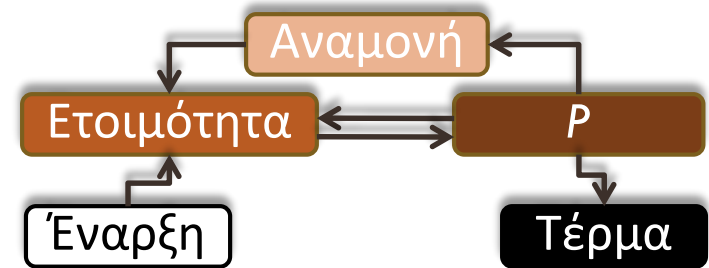
11 διεργασίες συνολικά, εκτός *Ετοιμότητας*

Άρα σε *Ετοιμότητα* βρίσκονται $16 - 11 = 5$ διεργασίες.

Χρονοπρογραμματισμός της ΚΜΕ

52

Ερώτηση: Όταν πρέπει να αντικατασταθεί η υπό εκτέλεση διεργασία, πώς το ΛΣ επιλέγει τον αντικαταστάτη της από αυτές που βρίσκονται σε *Ετοιμότητα*;



Απάντηση: Κάθε ΛΣ απαντά σε αυτό το ερώτημα με ένα σύνολο κανόνων για να προγραμματίζει ποια διεργασία θα εκτελεί κάθε φορά η ΚΜΕ.

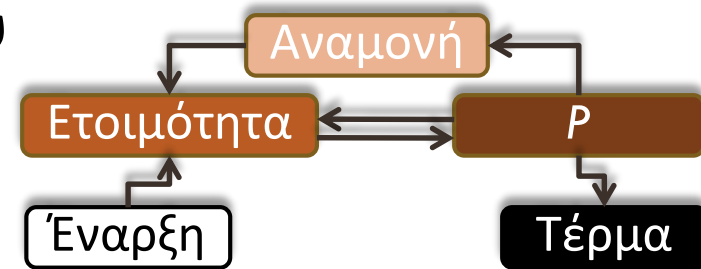
Λέμε ότι οι κανόνες αυτοί συνιστούν έναν **Αλγόριθμο Χρονοπρογραμματισμού της ΚΜΕ** (CPU scheduling algorithm).

Χρονοπρογραμματισμός της ΚΜΕ

53

Ο αλγόριθμος χρονοπρογραμματισμού καλείται όποτε ενδέχεται να αλλάξει η υπό Εκτέλεση διεργασία P , δηλαδή:

1. Όποτε η P Τερματίσει,
2. Όποτε η P τεθεί σε Αναμονή,
3. Όποτε η P εξαντλήσει τον χρόνο που της ανατέθηκε,
4. Όποτε κάποια διεργασία σε αναμονή τεθεί σε Ετοιμότητα (ίσως να έχει προτεραιότητα έναντι της P).



Στις περιπτώσεις 3 και 4, ο χρονοπρογραμματισμός που συντελείται λέγεται **Προεκτοπιστικός** (*Preemptive*). Στις 1 και 2, λέγεται **Μη Προεκτοπιστικός** (*Non-preemptive*).

Χρονοπρογραμματισμός της ΚΜΕ

54

Τρεις απλοί αλγόριθμοι χρονοπρογραμματισμού:

- 1. Κατά Σειρά Άφιξης (First-Come First-Served, FCFS):** Εφαρμόζεται μη προεκτοπιστικά και προτάσσει τη διεργασία που τέθηκε νωρίτερα σε ετοιμότητα.
- 2. Κατά Μικρότερο Χρόνο Εκτέλεσης (Shortest Job Next, SJN):** Εφαρμόζεται μη προεκτοπιστικά και προτάσσει τη διεργασία που θα απασχολήσει λιγότερο την ΚΜΕ.
- 3. Κυκλικά (Round Robin, RR):** Εφαρμόζεται προεκτοπιστικά και διατρέχει τις διεργασίες κυκλικά. Η ΚΜΕ εκτελεί κάθε διεργασία για το πολύ 1 **κβάντο χρόνου** (*time slice*) τη φορά. Αν αυτό εξαντληθεί, το ΛΣ επιστρέφει τη διεργασία σε κατάσταση Ετοιμότητας, και αρχίζει να Εκτελεί την επόμενη από τις διεργασίες σε Ετοιμότητα.

Χρονοπρογραμματισμός της ΚΜΕ

55

Ερώτηση: Έστω ότι οι διεργασίες p_1, p_2, p_3, p_4, p_5 τίθενται σε ετοιμότητα σχεδόν ταυτόχρονα (αλλά με την παραπάνω σειρά) και ότι θα απασχολήσουν την ΚΜΕ για όσες χρονικές μονάδες αναφέρει ο πίνακας:

p_1	p_2	p_3	p_4	p_5
140	75	320	280	125

Για καθέναν από τους αλγορίθμους *FCFS*, *SJN*, *RR*, ποιο είναι το **διάγραμμα Gantt** (*Gantt chart*) που θα προκύψει αν εφαρμόσουμε τον αλγόριθμο σε αυτές τις διεργασίες;

(Το διάγραμμα Gantt απεικονίζει ποια διεργασία εκτελεί η ΚΜΕ κάθε στιγμή ---βλ. επόμενες διαφάνειες)

Χρονοπρογραμματισμός της ΚΜΕ

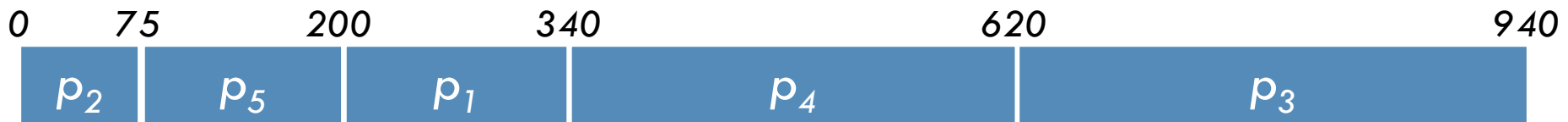
57

Απάντηση: Με τον SJN, οι διεργασίες «μπαίνουν» στην ΚΜΕ με την σειρά:

p_2, p_5, p_1, p_4, p_3

και την απασχολούν κατά το παρακάτω διάγραμμα Gantt:

p_1	p_2	p_3	p_4	p_5
140	75	320	280	125



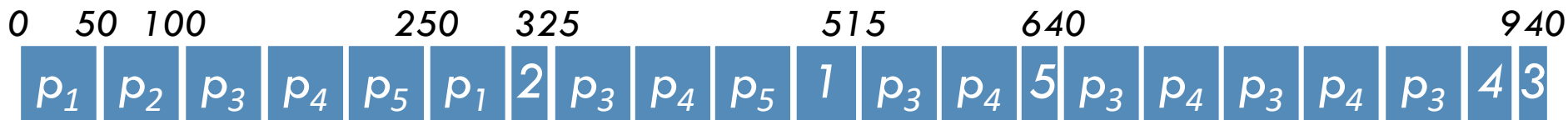
Χρονοπρογραμματισμός της ΚΜΕ

58

Απάντηση: Με τον *RR*, οι διεργασίες «μπαίνουν» στην ΚΜΕ με την σειρά:

$p_1, p_2, p_3, p_4, p_5, p_1, p_2, p_3, \dots$
και την απασχολούν κατά το παρακάτω διάγραμμα Gantt (έστω **κβάντο** 50 μονάδων):

p_1	p_2	p_3	p_4	p_5
140	75	320	280	125



Χρονοπρογραμματισμός της ΚΜΕ

59

Ερώτηση: Ποιος από τους αλγορίθμους είναι καλύτερος;

Απάντηση: Δεν υπάρχει μοναδικό κριτήριο.

Ένα από τα κριτήρια που χρησιμοποιούνται είναι ο **Μέσος Χρόνος Ολοκλήρωσης** (*average turnaround time*) επί όλων των διεργασιών.

Εδώ, **Χρόνος Ολοκλήρωσης** (*turnaround time*) μιας διεργασίας είναι ο χρόνος που μεσολαβεί από τη στιγμή που η διεργασία τίθεται σε ετοιμότητα για πρώτη φορά μέχρι τη στιγμή που παύει να εκτελείται για τελευταία φορά.

Χρονοπρογραμματισμός της ΚΜΕ

60

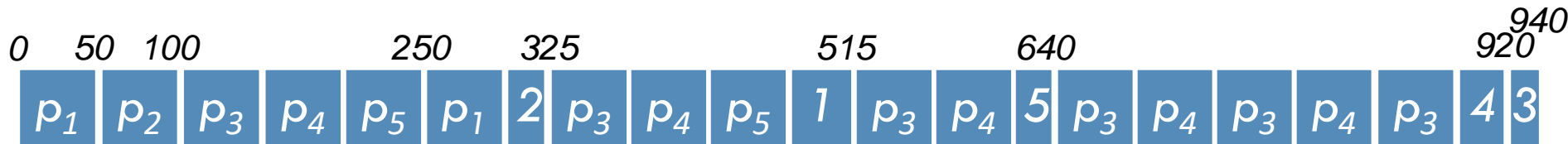
Ερώτηση: Αν όλες οι διεργασίες τέθηκαν ταυτόχρονα σε ετοιμότητα, ποιος είναι ο Μέσος Χρόνος Ολοκλήρωσης στα παρακάτω διαγράμματα Gantt;



Απάντηση: $(140+215+535+815+940)/5 = \mathbf{529}$



Απάντηση: $(75+200+340+620+940)/5 = \mathbf{435}$



Απάντηση: $(325+515+640+920+940)/5 = \mathbf{668}$

Χρονοπρογραμματισμός της ΚΜΕ

61

Επομένως, από την άποψη του Μέσου Χρόνου Ολοκλήρωσης, η αλγόριθμος *SJN* ήταν ο καλύτερος.

Όμως ο *SJN* έχει και μειονεκτήματα:

1. Δεν είναι πάντα δυνατόν να γνωρίζουμε την διάρκεια των διεργασιών.
2. Δεν αποκλείει το φαινόμενο της **Στέρησης** (*starvation*). (Πρόκειται για το φαινόμενο κατά το οποίο μια διεργασία δεν εκτελείται ποτέ διότι διαρκώς καταφθάνουν και τίθενται σε ετοιμότητα άλλες διεργασίες, με μικρότερη διάρκεια.)

Ερωτήσεις;